

GPIO

gpioControl.py

```
import RPi.GPIO as GPIO
from time import sleep

ina1 = 33
ina2 = 35
ena = 37

inb1 = 31
inb2 = 29
enb = 23

GPIO.setmode(GPIO.BOARD)

GPIO.setup(12, GPIO.OUT, initial=1)
GPIO.setup(18, GPIO.OUT, initial=1)

GPIO.output(12, GPIO.LOW)
GPIO.output(18, GPIO.LOW)
GPIO.setup(ina1, GPIO.OUT)
GPIO.setup(ina2, GPIO.OUT)
GPIO.setup(ena, GPIO.OUT)
GPIO.setup(inb1, GPIO.OUT)
GPIO.setup(inb2, GPIO.OUT)
GPIO.setup(enb, GPIO.OUT)

p1 = GPIO.PWM(18, 50) # 50 Hz
p2 = GPIO.PWM(12, 50) # 50
pa = GPIO.PWM(ena, 1000)
pb = GPIO.PWM(enb, 1000)

cameraInitPositionY = 6.0
cameraInitPositionX = 6.5

verticalVal = cameraInitPositionY
horizontalVal = cameraInitPositionX

class GpioControl(object):

    def __init__(self):
        global verticalVal
```

```

global horizontalVal
global p1
global p2
p1.start(0)
p2.start(0)
pa.start(25)
pb.start(25)
p1.ChangeDutyCycle(cameraInitPositionY)
p2.ChangeDutyCycle(cameraInitPositionX)
sleep(0.1)
p1.ChangeDutyCycle(0)
p2.ChangeDutyCycle(0)
verticalVal = cameraInitPositionY
horizontalVal = cameraInitPositionX
#print("> Init Vert=" + str(verticalVal) + ",Hort=" + str(horizontalVal))

def __del__(self):
    global p1
    global p2
    p1.stop()
    p2.stop()
    print(" GPIO.__del__() ")
    GPIO.cleanup()

def cleanUp(self):
    global p1
    global p2
    p1.stop()
    p2.stop()
    print(" GPIO.cleanUp() ")
    GPIO.cleanup()
    sleep(2)

def initMotorPosition(self):
    # Init
    global verticalVal
    global horizontalVal
    global cameraInitPositionY
    global cameraInitPositionX
    global p1
    global p2
    p1.ChangeDutyCycle(float(cameraInitPositionY))
    p2.ChangeDutyCycle(float(cameraInitPositionX))
    sleep(0.1)
    p1.ChangeDutyCycle(float(0.0))
    p2.ChangeDutyCycle(float(0.0))
    verticalVal = cameraInitPositionY
    horizontalVal = cameraInitPositionX
    #print("> Init Vert=" + str(verticalVal) + ",Hort=" + str(horizontalVal))
    return "Vert=" + str(verticalVal) + ",Hort=" + str(horizontalVal)

```

```

def moveUp(self):
    global verticalVal
    global horizontalVal
    global p2
    if(verticalVal>3.5):
        verticalVal = float(round(verticalVal-0.2, 1))
        p2.ChangeDutyCycle(verticalVal)
        #print("> UP Vert=%.2f ,Hort=%.2f" % (verticalVal, horizontalVal))
        sleep(0.1)
        p2.ChangeDutyCycle(0)
    return "UP Vert=" + str(verticalVal) + ",Hort=" + str(horizontalVal)

def moveDown(self):
    global verticalVal
    global horizontalVal
    global p2
    if(verticalVal<8.1):
        verticalVal = float(round(verticalVal+0.2, 1))
        p2.ChangeDutyCycle(verticalVal)
        #print("> Down Vert=" + str(verticalVal) + ",Hort=" + str(horizontalVal))
        sleep(0.1)
        p2.ChangeDutyCycle(0)
    return "DOWN Vert=" + str(verticalVal) + ",Hort=" + str(horizontalVal)

def moveRight(self):
    global verticalVal
    global horizontalVal
    global p1
    if(horizontalVal>2.5):
        horizontalVal = float(round(horizontalVal-0.2, 1))
        p1.ChangeDutyCycle(horizontalVal)
        #print("> Right Vert=" + str(verticalVal) + ",Hort=" + str(horizontalVal))
        sleep(0.1)
        p1.ChangeDutyCycle(0)
    return "RIGHT Vert=" + str(verticalVal) + ",Hort=" + str(horizontalVal)

def moveLeft(self):
    global verticalVal
    global horizontalVal
    global p1
    if(horizontalVal<10):
        horizontalVal = float(round(horizontalVal+0.2, 1))
        p1.ChangeDutyCycle(horizontalVal)
        #print("> Left Vert=" + str(verticalVal) + ",Hort=" + str(horizontalVal))
        sleep(0.1)
        p1.ChangeDutyCycle(0)
    return "LEFT Vert=" + str(verticalVal) + ",Hort=" + str(horizontalVal)

```

```
def goForward(self,time):
    print("Go forward")
    pa.ChangeDutyCycle(75)
    GPIO.output(ina1,GPIO.HIGH)
    GPIO.output(ina2,GPIO.LOW)
    pb.ChangeDutyCycle(75)
    GPIO.output(inb1,GPIO.HIGH)
    GPIO.output(inb2,GPIO.LOW)
    sleep(0.1*int(time))
    GPIO.output(ina1,GPIO.LOW)
    GPIO.output(ina2,GPIO.LOW)
    GPIO.output(inb1,GPIO.LOW)
    GPIO.output(inb2,GPIO.LOW)
    return " >> Go Forward"

def goBackward(self,time):
    print("Go backward")
    pa.ChangeDutyCycle(75)
    GPIO.output(ina1,GPIO.LOW)
    GPIO.output(ina2,GPIO.HIGH)
    pb.ChangeDutyCycle(75)
    GPIO.output(inb1,GPIO.LOW)
    GPIO.output(inb2,GPIO.HIGH)
    sleep(0.1*int(time))
    GPIO.output(ina1,GPIO.LOW)
    GPIO.output(ina2,GPIO.LOW)
    GPIO.output(inb1,GPIO.LOW)
    GPIO.output(inb2,GPIO.LOW)
    return " >> Go Backward"

def goTurnleftback(self,time):
    print("Go Turn Left back")
    pa.ChangeDutyCycle(75)
    GPIO.output(inb1,GPIO.LOW)
    GPIO.output(inb2,GPIO.HIGH)
    sleep(0.1*int(time))
    GPIO.output(ina1,GPIO.LOW)
    GPIO.output(ina2,GPIO.LOW)
    GPIO.output(inb1,GPIO.LOW)
    GPIO.output(inb2,GPIO.LOW)
    return " >> Go Turn Left back"

def goTurnrightback(self,time):
    print("Go Turn Right back")
    pb.ChangeDutyCycle(75)
    GPIO.output(ina1,GPIO.LOW)
    GPIO.output(ina2,GPIO.HIGH)
    sleep(0.1*int(time))
    GPIO.output(ina1,GPIO.LOW)
    GPIO.output(ina2,GPIO.LOW)
```

```
GPIO.output(inb1,GPIO.LOW)
GPIO.output(inb2,GPIO.LOW)
return " >> Go Turn Right back"

def goTurnleftforward(self,time):
    print("Go Turn Left forward")
    pa.ChangeDutyCycle(75)
    GPIO.output(inb1,GPIO.HIGH)
    GPIO.output(inb2,GPIO.LOW)
    sleep(0.1*int(time))
    GPIO.output(ina1,GPIO.LOW)
    GPIO.output(ina2,GPIO.LOW)
    GPIO.output(inb1,GPIO.LOW)
    GPIO.output(inb2,GPIO.LOW)
    return " >> Go Turn Left forward"

def goTurnrightforward(self,time):
    print("Go Turn Right forward")
    pb.ChangeDutyCycle(75)
    GPIO.output(ina1,GPIO.HIGH)
    GPIO.output(ina2,GPIO.LOW)
    sleep(0.1*int(time))
    GPIO.output(ina1,GPIO.LOW)
    GPIO.output(ina2,GPIO.LOW)
    GPIO.output(inb1,GPIO.LOW)
    GPIO.output(inb2,GPIO.LOW)
    return " >> Go Turn Right forward"
```

🕒Revision #1

★Created 2023-06-01 14:40:29 UTC by Hyeon Su Ryu

✎Updated 2023-06-01 14:41:16 UTC by Hyeon Su Ryu